Towards one Symbol Network Coding Vectors

Nikolaos Thomos, *Member, IEEE*, and Pascal Frossard, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a novel design for network coding vectors that limits the overhead information. Network coding vectors contain information regarding the operations the packets have undergone in the network nodes. They are used at the decoder side to invert coding operations and recover the data. We propose to reduce the size of this side information with the use of Vandermonde-like generator matrices at the sources. These matrices permit to describe the coding operations performed on packets with only one symbol. We analytically investigate the limitations arising from such design constraints. Interestingly, we find that the feasible generation size is upper bounded by $\log_2 q$ in Galois field $\mathbb{F}_q$ of size q as this is the maximum packet diversity allowed by the employed generator matrices. In addition, we show that network coding nodes should only perform addition operations in order to maintain the properties of the coding vectors. We finally discuss the benefits and limitations of the proposed coding vectors in practical systems.**

***Index Terms*— Network coding, coding vectors, header compression.**

I. Introduction

Network coding [1], [2] has attracted a lot of attention during the past decade since it promises improved network throughput, decreased delivery delay, and decentralized control, among other interesting properties. One of the first attempts to make network coding practical has been presented in [3] where Randomized Linear Network Coding (RLNC) [4] is employed. RLNC is pretty simple and does not require coordination between network nodes, which has made it popular in practical systems. It however requires each packet to be augmented with a network coding header that contains information about the coding operations in the network. In order to keep the size of the network coding header reasonable, the network coding operations are limited to groups of packets sharing similar decoding deadlines. These groups of packets are known as generations. Then, the length of a network coding vector is $N \log_2 q$ bits, where N denotes the number of packets in a generation and q the size of the employed Galois field $\mathbb{F}_q$ where the coding operations are performed. Although, the segmentation in generations shortens the length of network coding vectors, the related overhead might still affect the goodput of the system.

In this paper, we propose a design that shortens the employed network coding vectors of RLNC schemes by imposing a specific design of the generator matrices. Our work is influenced by Rateless codes [5], [6] where a seed of a pseudo-random generator is used to determine which source symbols have been combined for the generation of a rateless encoded symbol. Therefore, if one initializes the pseudo-random generator with a given seed, it produces always the same sequence of numbers. Similarly, in RLNC systems a naive approach could be to employ a seed of a pseudo-random generator that produces a sequence of network coding coefficients. Although this method is efficient in terms of compression, it necessitates all the network nodes to be synchronized. This is a limiting factor in the deployment of distributed RLNC systems. Another drawback of such an approach is the need for large look-up tables which leads to high computational complexity. Instead, we employ a form of modified Vandermonde matrices for the generation of coding coefficients, which could be then determined uniquely by one single seed symbol. In addition, the coding operations in the network nodes results in the generation of another coding vectors that is also uniquely described by another seed symbol. The analysis of the proposed design which applies linear coding in $\mathbb{F}_q$ at sources and coding operations in $\mathbb{F}_2$ in the network nodes, shows that our method constitutes a tradeoff between small header overhead, and the goodput of the network coding system. Indeed, the set of valid packet combinations is limited by the design of the coding coefficient vectors. This may limit the throughput benefits in the network coding system. We also show that the generation size should not exceed $\log_2 q$ due to the cyclic property of the Vandermonde matrices. Finally, we discuss methods for the deployment of the proposed network coding solution in practical settings.

The works in [7] and [8] are probably the studies that are the closest to our proposed system. The first attempt to reduce the size of the transmitted network coding vector without any performance loss has been made in [7]. This design has been proposed for RLNC [4]. The sources do not combine all the packets and a unique vector of the form $[0\ 0\ 1\ \ldots\ 0]$ is appended to each packet at servers. Without loss of generality, as the packets travel through the network, only m of them are combined in network nodes. The resulting coding vector is compressible to a vector of size $2m$. This becomes feasible by coding with parity check matrices like those used by channel codes. The performance of this compression scheme has been improved in [8]. A segment ID is added to the header in order to record the IDs of the sources that have been combined in a packet. The achievable compression is $m + n/\log q$, where n stands for the generation size. The compression comes arbitrarily close to $m + O(\log n)/\log q$ [8] when a list decoding scheme is employed. In contrary to the works in [7] and [8] we do not assume sparse coding vectors. We focus on RLNC that operates on dense vectors, which permits to exploit better the network resources. It is however critical to reduce

The manuscript received July 31 2012. The associate editor coordinating the review of this letter and approving it for publication was M. Leeson.

The authors are with Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, Signal Processing Laboratory, Laussane, Switzerland (e-mail: nikolaos.thomos@epfl.ch; pascal.frossard@epfl.ch).

Nikolaos Thomos is also with Communication and Distributed Systems laboratory (CDS), University of Bern, Bern, Switzerland.

This work has been supported by the Swiss National Science Foundation, under grants PZ00P2-121906 and PZ00P2-137275.

the amount of bandwidth spent for sending the coding vectors, as the coding vector overhead can become comparable to the payload.

II. Network coding vector design

Coding coefficient vectors employed by RLNC schemes typically consume significant bandwidth resources. The reduction of the bandwidth needed for communicating the coding vectors leads to more efficient exploitation of the network resources. Ideally, network coding vectors should be of minimal size, *i.e.*, one symbol. At the same time, the linear properties of RLNC schemes should be maintained, *i.e.*, combinations of coding vectors should result in another valid short coding vector, such that decoding can be achieved by classical algorithms like Gaussian Elimination.

In order to achieve these properties, we propose to design coding vectors based on Vandermonde matrices. These matrices have the following structure

$$\mathbf{A}^{'} = \begin{bmatrix} 1 & a_1 & a_1^2 & a_1^3 & \dots & a_1^n \\ 1 & a_2 & a_2^2 & a_2^3 & \dots & a_1^n \\ \vdots & & & \ddots & \vdots & \\ 1 & a_m & a_m^2 & a_m^3 & \dots & a_m^n \end{bmatrix}$$

Assuming that each row of $\mathbf{A}^{'}$ represents a coding vector, we observe that the elements in the second column of $\mathbf{A}^{'}$ could be seen as "seeds" of a pseudo-random generator, *i.e.*, one can produce all other entries of a row using only one of these seeds. We adopt modified Vandermonde matrices that do not contain the first column. In particular, we want that the addition in $\mathbb{F}_q$ of coding vectors (rows of the matrix) is uniquely described by a seed representing in another row of the matrix. This is however not always true in the above matrix, as

$$\mathbf{a}_1 + \mathbf{a}_2 \neq \mathbf{a}_i, \; i \in [1, m] \text{ and } i \neq 1, 2,$$

where $\mathbf{a}_i, \mathbf{a}_1, \mathbf{a}_2$ are vectors of the form $[a \; a^2 \; \dots \; a^n]$. But this condition becomes true when the coding vectors are of the form $[a_i \; a_i^2 \; a_i^{2^2} \; \dots \; a_i^{2^n}]$ except for $a_i = 1$. This interestingly resembles the design of parity check matrices used for maximum rank distance (MRD) codes [9] such as Gabidulin codes [10]. Hence, we propose to employ generator matrices A of the form

$$\mathbf{A} = \begin{bmatrix} a_1 & a_1^{[2]} & a_1^{[3]} & \dots & a_1^{[n]} \\ a_2 & a_2^{[2]} & a_2^{[3]} & \dots & a_1^{[n]} \\ \vdots & & \ddots & \vdots & \\ a_m & a_m^{[2]} & a_m^{[3]} & \dots & a_m^{[n]} \end{bmatrix}$$

where $a_i^{[j]} = a_i^{2^j}$.

Due to the cyclic property of the Galois fields, $\mathbf{A}$ has a periodicity of $M = \log_2 q$ symbols in every row. Thus, we have $a_i^{[j]} = a_i^{[j+M]}$. The addition of columns of $\mathbf{A}$ forms a valid codeword. This can be understood from the linearity constraint of the MRD codes, which are similar to our codes.

In RLNC based systems, the packets are also subject to multiplication by linear combinations in the network nodes. Under multiplication of a vector $\mathbf{a}_1$ with a scalar γ, we have

$$[a_1\gamma \;\; a_1^{[2]}\gamma \;\; a_1^{[3]}\gamma \;\; \dots \;\; a_1^{[M]}\gamma] = [a_i \; a_i^{[2]} \; a_i^{[3]} \; \dots \; a_i^{[M]}]$$

Thus, we have a set of equation of the form

$$a_1^{[i]}\gamma = a_i^{[i]}, \quad i = 1, \dots, M$$

The only non-trivial scalar value that simultaneously satisfies all the above equalities is $\gamma = 1$. Such a choice however results in the original coding vector. Hence, multiplication in network nodes are not compatible with our design of the coding coefficient vectors.

To summarize, two constraints are associated with the design of network coding vectors: (a) the generation size should not exceed $M = \log_2 q$ because of the cyclic property of the employed generator matrices at sources and (b) only additions between the network coding vectors are permitted.

III. Design analysis

In order to constrain the size of the coding coefficient vector, our design limits the set of possible coding operations in the network and hence the diversity of the packets.

Since every network coding vector is characterized by the first element of the corresponding row of $\mathbf{A}$ (seed), its choice limits the number of possible coding operations. In this section, we focus on the probability to randomly generate at sources an innovative packet, *i.e.*, to produce novel information with respect to the packets that have already been generated. The analysis of this probability brings information about the performance penalty induced by our constrained encoding design.

We first examine the probability for the m-th network coded packet to be innovative. We also determine the size of the set of eligible seeds that permit the construction of full rank equation systems at the decoder side. For the first packet, we know that it is non-innovative when the seed "0" (corresponds to all zeros coding vector) is selected as we lose all the source information. Seed "1" (corresponds to all ones coding vector) should also not be selected as the corresponding coding vector does not result in a valid seed under addition. If we denote with $\mathcal{S}(1)$ the set of the valid seeds when the first coding vector is selected at sources, we have $\mathcal{S}(1) = \mathbb{F}_q \backslash \{0, 1\}$. Thus, the probability of selecting a non-innovative[1] packet is

$$\overline{P}_{inv}(1) = \frac{q - |\mathcal{S}(1)|}{q}$$

Thus, it is $\overline{P}_{inv}(1) = 2/q$. For the second packet we have $\mathcal{S}(2) = \mathbb{F}_q \backslash \{0, 1, a_1\}$ as the selected seed should also differ from that of the first packet. Trivially, it is $\overline{P}_{inv}(2) = 3/q$. In similar way, for the third packet we have that $\mathcal{S}(3) = \mathbb{F}_q \backslash \{0, 1, a_1, a_2, a_1 + a_2\}$ as the seed of the third coding vector should not correspond to any of the previously selected seeds nor their addition. Similarly for the m-th coding vector, we have

$$q - \mathcal{S}(m) = 2 + (m-1) + \sum_{i=2}^{m-1} \binom{m-1}{i}$$

[1]Please note that the term non-innovative includes the vectors of the form $[1 \; 1 \; \dots \; 1]$ that are described by the seed "1" as non innovative since they do not satisfy the design constraints.

The probability that the m-th packet is non innovative is equal to

$$\overline{P}_{inv}(m) = \frac{q - |\mathcal{S}(m)|}{q}$$

Therefore, the probability that all the m packets that are transmitted from the source are innovative is

$$P_{inv} = \prod_{i=1}^{m}(1 - \overline{P}_{inv}(i)) \tag{1}$$

Finally, the number of valid coding vectors for the selection of the seed of the m-th coding vector is $q - |\mathcal{S}(m)|$.

From the above, we can observe that, for $m > q$, we have $\mathcal{S}(m) > |\mathbb{F}_q|$. Thus, we cannot employ generations of symbols that are larger than $\log_2 q$. Moreover, for a generation of size $m < q$, the probability that the randomly selected seeds result in rank deficient system is $1 - P_{inv}$. We should note that in our system, the servers select the packet seeds taking into account the previously selected seeds. Intermediate nodes perform operations with the received packets by means of addition. We would like to emphasize that we employ shorter headers than [7], [8] and the length of the employed header does not augment with the number of processing operations (combinations) in the network, contrarily to [7], [8].

Finally, we would like to note that we focus only on the probability of generating innovative packets at the sources, as it is the main source of performance degradation of our scheme. Some performance degradation is introduced by the coding operations in the intermediate nodes. This is due to the fact that the addition of two or more packets might result in the same network coding header as another packet that is available in the node's buffer. However, the same type of problems arises in other RLNC coding schemes, even if the actual penalty depends on the packet types and network settings. The main difference between our scheme and other RLNC schemes is thus due to the coding strategy at the sources.

IV. Performance analysis

In the previous section, we have determined the performance limits of the proposed network coding algorithm, which minimizes the length of the coding vectors. We now illustrate the performance of our algorithm by simulations. First, we investigate the probability that a randomly chosen coding vector is innovative with respect to the number of seeds of the coding vectors that have been already generated. We present results coming from the evaluation of Eq. (1) for two coding field sizes, $\mathbb{F}_{2^8}$ and $\mathbb{F}_{2^{15}}$. From Fig. 1, we can observe that for the first few selections of seeds the probability that a randomly selected packet is innovative is approaching 1. We also see that, as the number of packets increases, the corresponding probability decreases sharply. As expected, the probability of finding innovative packets decreases smoothly with the size of the Galois field. The results are presented for a number of packets that does not exceed $M = 8$ and $M = 15$ for $\mathbb{F}_{2^8}$ and $\mathbb{F}_{2^{15}}$ respectively, as we cannot have more innovative packets, by design.

The same conclusions can be drawn if we examine the proposed scheme in terms of number of available eligible seeds

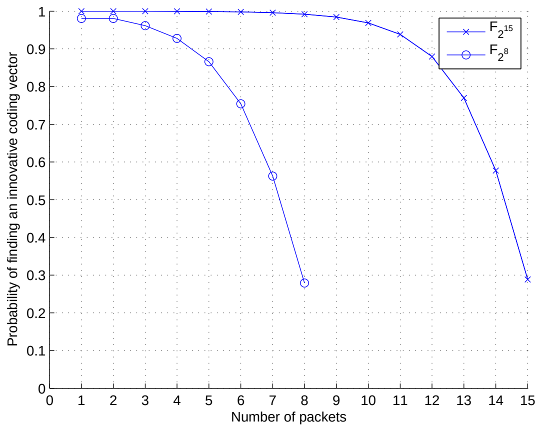
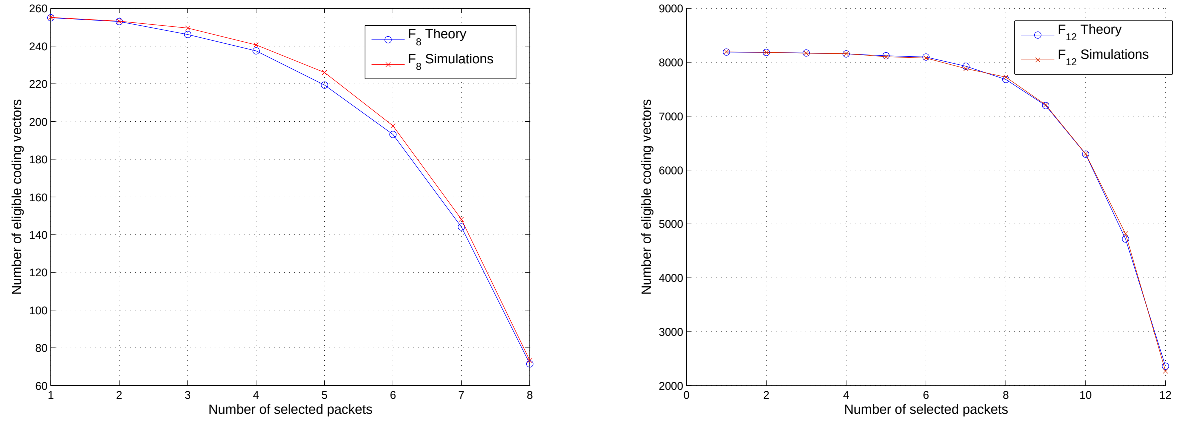


Fig. 1: Probability of generating an innovative coding vector at sources with respect to the number of network coded packets for $\mathbb{F}_{2^8}$ and $\mathbb{F}_{2^{15}}$.

for coding vectors with respect to the number of randomly selected packets. The theoretical results are compared with simulation results. From Figs. 2(a) and (b), we can see that the simulation results match the analysis and that only marginal differences are observed. We can also note that the number of available seeds decreases when the number of generated packets increases. Furthermore, it decreases fast as the number of generated packets approaches M. It is also obvious that, when our system employs larger finite field sizes, it has a lower probability of generating non-innovative packets.

For the sake of completeness, we have investigated the performance of the proposed scheme in terms of the time required to collect enough packets for decoding. The results are illustrated in Fig. 3. These resullts also reveal the goodput gains, as higher goodput is associated with faster reception of packets. We compare the proposed scheme with RLNC in regular networks consisting of three nodes per coding stage, where every node is connected with all nodes in the previous stage [11].[2] The number of coding stages varies from two to four. All the links have capacity equal to 200 symbols/sec (1 symbol = 12 bits). In our scheme, we restrict the coding operations to pairs of symbols in order to increase the packet diversity and to decrease the probability that some information is eliminated due to random coding operations. The operations are performed in $\mathbb{F}_{2^{12}}$. From Section III, we have seen that the generation size is limited to 12 in this case. Hence, in order to maintain a high probability of innovative packets, we set the generation size equal to 10. We consider packets with small payloads (50 and 90 bytes), which are typical in sensor networks. In the proposed setting, the sources transmit packets until decoding is possible to all clients. We see in Fig. 3 that larger gains are noticed for smaller payloads due to lower overhead in this case. Under the same bandwidth conditions, the proposed method permits to receive the packets

[2]We omit the performance comparisons with optimized routing solutions and rather use RLNC as a baseline scheme. In general, routing solutions are inferior to RLNC schemes operating in large Galois fields as RLNC schemes can approach the min-cut of the underlying communication graph. However, the gains of the network coding approaches depend on the network topology.



(a) (b)

Fig. 2: Number of eligible coding vectors at sources with respect to the number of network coded packets for: (a) $\mathbb{F}_{2^8}$ and (b) $\mathbb{F}_{2^{12}}$.

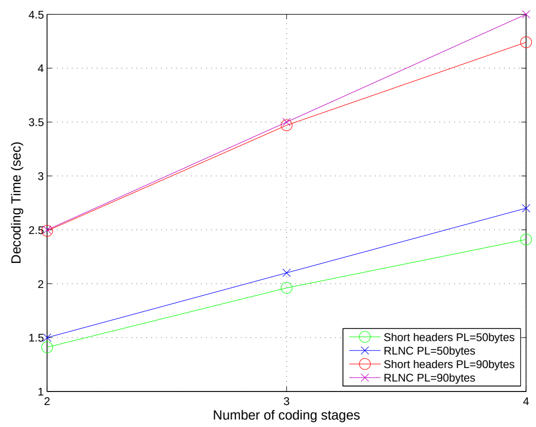


Fig. 3: Decoding times with respect to the number of coding stages. The proposed methods "Short headers" is compared with RLNC for payloads of 50 and 90 bytes, while the generation size is 10 and operations are performed in $\mathbb{F}_{2^{12}}$.

faster as for every packet transmission the header is nine symbols smaller. Further, we observe that the performance difference increases with the number of coding stages. This is due to the fact with smaller coding vectors each intermediate node receives packets faster, thus it is able to send also faster. However, we should note that with the proposed method we have higher probability of receiving non-innovative packets as coding operations in intermediate nodes are performed between pairs of packets. This leads to performance penalty for large networks as the probability of generating redundant packets increases with the number of coding stages.

V. Discussion

We have presented a novel design of network coding vectors that permits their representation with one symbol. The proposed method is particularly interesting for networks where the communicated data has small payload and thus the use of large coding vectors is prohibited. We find that the employed generation size is upper bounded by $M = \log_2 q$. Our system is also appropriate for low-cost intermediate nodes as operations only consist of modulo-2 additions. However, for guaranteeing the maximum packet diversity at the network, sources should carefully select the coding vectors of the packets. Finally, the proposed method offers gains in terms of the time required for decoding compared to traditional RLNC approaches that employ large coding vectors compared to the packet payload. The proposed approach is particularly useful in sensor networks, since sensors have readings (messages) with small payload and thus data should be packetized in few packets (small generations).

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Trans. Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[3] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. 41st Allerton Conf. on Communication Control and Computing*, Monticell, IL, USA, Oct. 2003.

[4] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On Randomized Network Coding," in *41st Allerton Annual Conference on Communication, Control, and Computing*, Monticello, IL, USA, Oct 2003.

[5] M. Luby, "LT codes," in *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '02)*, Vancouver, Canada, Nov. 2002, pp. 271–280.

[6] A. Shokrollahi, "Raptor codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[7] M. J. Siavoshani, L. Keller, K. Argyraki, and C. Fragouli, "Compressed Network Coding Vectors," in *Proc. of Int. Symp. on Information Theory,ISIT'09*, Seoul, South Korea, June 2009.

[8] S. Li and A. Ramamoorthy, "Improved Compression of Network Coding Vectors Using Erasure Decoding and List Decoding," *IEEE Communications Letters*, vol. 14, no. 8, pp. 749–751, Aug. 2010.

[9] D. Silva, F. R. Kschischange, and R. Kotter, "A Rank-Metric Approach to Error Control in Random Network Coding," *IEEE Trans. Information Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.

[10] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Rank Errors and Rank Erasures Correction," in *Proc. 4th Int. Colloq. Coding Theory*, Dilijan, Armenia, 1992.

[11] N. Thomos and P. Frossard, "Network Coding of Rateless Video in Streaming overlays," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1834–1847, Dec. 2010.